Remarks

In the Office Action dated July 18, 2005 ("Office Action"), Claims 1-22 were rejected. All Claims remain as previously presented. In view of the comments set forth below, it is respectfully submitted that all Claims are in condition for allowance.

1. Applicants' acknowledge the duty to disclose information under 37 CFR 1.56.

2. Page 1 of the specification has been updated above to include status information and the applicable serial numbers.

3. Claims 1, 6-8, and 19-22 were rejected under 35 USC §102(a) as being anticipated by pages 1-3 of Applicants' Admitted Prior Art ("AAPA") pages 1-3. This rejection is respectfully traversed.

Before considering the language of the Claims in detail, a summary of Applicants' invention is provided for discussion purposes. Applicants' invention provides a system and method for maintaining memory coherency within a multiprocessor environment that includes multiple requesters such as Instruction Processors (IPs) that are coupled to a shared main memory. Within this environment, each IP may initiate retrieval of data from the memory. This retrieved data may be stored within one or more caches accessible to that IP.

When a processor retrieves data from the memory for update purposes, other read-only copies of the data that may be stored elsewhere within the system must be invalidated. According to an acceleration mechanism within one embodiment of the system, data may be provided to an IP for update purposes before all other read-only copies of this data stored within other caches in the system have been invalidated.

To ensure that the acceleration mechanism described above does not result in memory incoherency, a novel instruction is provided for inclusion within the hardware instruction set of an IP. This instruction, which in one embodiment is an "ENZ instruction", is executed to cause the IP that is updating the data to stall until all outstanding invalidation activities have completed. Most notably, this instruction will cause the IP to stall until outstanding invalidation activities have completed for the data this is being updated. When the IP completes execution of the ENZ instruction such that the IP is no longer stalled, the IP may then safely initiate the transfer of the updated data back to main memory without causing any memory incoherency.

The novel ENZ instruction may be executed by an IP that updated data so that this IP can ensure that all outdated copies of this data have been invalidated prior to issuing an Inter-Processor Interrupt (IPI) that informs other IPs within the system that the updated data is available for use. If the outdated data copies were not invalidated before this type of IPI is issued, some of those other IPs may utilize the outdated data copies. This results in memory coherency. (See, for example, pages 3-6 and 35-46 of Applicants' Specification.)

Before continuing, it may be noted that the foregoing summary is intended to be entirely consistent with that shown in Applicants' Figures and described in the Specification.

Next, the language of Claim 1 is considered. This Claim appears as follows:

> 1. For use in a data processing system having a memory coupled to multiple requesters, a memory coherency system, comprising:
>
> a memory circuit coupled to provide a copy of requested data from the memory to a first requester, and to initiate invalidation operations to invalidate all read-only copies of the requested data that are stored by one or more other requesters; and

> a circuit included within the first requester and responsively coupled to the memory circuit to execute an instruction that causes the first requester to temporarily enter a stalled state until all of the invalidation operations have been completed. (Claim 1, emphasis added)

This Claim, as originally presented, describes a circuit to execute an instruction, which in one embodiment is the ENZ instruction discussed above. Execution of this instruction by the circuit causes the first requester, which in one embodiment is an IP, to temporarily enter a stalled state until all of the invalidation operations initiated to invalidate read-only copies of the requested data have been completed. The circuit that executes the instruction and provides this functionality is shown in Applicants' Figures 2 and 3 and is described in the associated portion of the Specification.

The Examiner states that this circuit is taught by Applicants' Admitted Prior Art (AAPA). In particular, the Examiner cites lines 23 and 29-30 of page 2, and lines 5-8 of page 3 as teaching a circuit for execution of an instruction that causes the first requester to temporarily enter a stalled state until all of the invalidation operations have been completed. (See Office Action page 3, lines 7-11.)

The relevant portion of the Specification is set forth below, with specific the passages cited by the Examiner being underlined, as follows:

> "One of the ways cache coherency is maintained is by invalidating old copies of data before data is provided to an IP. For example, assume a first IP requests a copy of data for update purposes, and another read-only copy of the data exists within the cache of a second IP. The read-only copy must generally be invalidated before a copy of this data is provided to the first IP. This ensures that an old copy of data is not used after the first IP has updated its new copy. One problem with this methodology is that is may take some

time to invalidate the read-only copy, delaying processing activities of the first IP." (Specification page 2 lines 23-30.)

This paragraph describes a first prior art architecture for maintaining cache coherency. According to this prior art architecture, all old copies of the data must be invalidated before the memory will provide data to the IP. This paragraph states the obvious: because data is not provided to the IP until coherency operations are completed, the IP may have to wait to begin processing this data. However, once the IP receives the data, the IP "knows" the invalidation operations are completed.

The cited portion of this passage, and indeed the paragraph as a whole, has absolutely nothing to do with a circuit included in the requester (e.g., an IP) for executing an instruction that causes the stalling of that requester until invalidation operations are complete. In fact, it may be noted that in the type of system described in this paragraph, there is no need for this type of circuit within the IP. That is because the memory guarantees that all invalidation activities are completed before the data is ever provided to the requester. (See first three lines of this paragraph.) Thus, it is completely unnecessary, and in fact would unnecessarily delay operation, if the IP executed any instruction to ensure that invalidation operations had been completed.

To re-state, the cited passage on page 2 of Applicants' Specification most certainly does not in any way teach, or even vaguely suggest, any type of circuit within a requester for executing an instruction that causes stalling of the requester until invalidation operations are complete. Such a circuit would only unnecessarily diminish throughput if used in an architecture of the type described in this passage. If this rejection is maintained, clarification regarding the perceived teaching of the cited passage on page 2 of Applicants' Specification is respectfully requested.

Next, the paragraph on page 3 of Applicants' Specification that contains the second cited passage is considered, as follows (with the cited portions being underlined):

> "An alternative to imposing the foregoing limitation involves providing data to a requester such as an IP before older data copies have been invalidated. Some prior art systems have allowed this to occur while also maintaining coherency by utilizing a single memory channel to enforce ordered request and response processing. In these types of systems, a restriction is imposed whereby a memory request to invalidate a read-only data copy must be completed before the memory provides any data in response to a subsequently issued memory request. This restriction guarantees memory coherency is maintained. The request/response ordering is accomplished by virtue of the fact that the memory includes a single memory channel to handle both memory requests and memory responses." (Specification page 3 lines 1-10)

Before continuing, it is important to note that this passage is describing a completely different architecture as compared to the first prior art architecture considered above in the first cited passage. According to this second prior art system, an acceleration mechanism is provided to allow requested data to be provided to a requester (e.g., an IP) before invalidation operations associated with the requested data have been completed. Recall that this type of operation was not allowed in the first prior art architecture considered above. The second prior art architecture is able to allow this acceleration mechanism to be used because this system also utilizes a single-channel memory design which enforces the ordering of requests and responses. This single-channel design guarantees that a memory request to invalidate an out-dated read-only data copy will be completed before the memory honors any subsequent request, thereby ensuring the memory remains coherent.

The passage of page 3 set forth above, and indeed the entire paragraph that relates to this second prior art architecture, has nothing whatsoever to do

with any type of circuit that is included in the requester (e.g., an IP) for executing an instruction that causes the stalling of that requester until invalidation operations are complete. In fact, in this second prior art architecture, there is absolutely no need to stall processor activity to maintain coherency, because coherency is maintained by virtue of the single-channel memory design. Thus, stalling the processor in this type of system would only serve to impose unnecessary delay, as was the case with the first prior art architecture. It is not understood how this passage is perceived to teach, or even vaguely suggest, the invention of Applicants' Claim 1. If this rejection is maintained, clarification regarding the perceived significance of this passage is requested.

Finally, it should be noted that the Examiner's rejection is tantamount to an improper obviousness-type rejection. The Examiner is citing portions of a description related to a first prior art architecture, and is further citing different portions of another description related to an entirely different second prior art architecture. These two architectures are diametrically opposed: the first architecture will only operate correctly if data is provided to a requester after all invalidation operations are completed, and the second architecture is premised on providing data before invalidation operations are completed. Whereas the first architecture does not require restrictions on the memory design, the second architecture requires use of a single-channel memory design. Thus, the two descriptions of these two different prior art architectures are mutually exclusive, and teach away from one another. The Examiner cannot "mix-and-match" portions of the description relating to the first prior art architecture with other portions of the description relating to the second prior art architecture in attempt to piece together the claimed invention of Claim 1. "Mixing-and-matching" in this manner will result in a non-operational system, and at any rate, most certainly does not teach Applicants' claimed invention. For this additional reason, this rejection is improper, and should be withdrawn.

To summarize, nothing in the AAPA describes, teaches, or even vaguely suggests, a circuit included within a requester to execute an instruction that causes that requester to enter a stalled state until all of the invalidation operations have been completed. If this rejection is maintained, a more detailed discussion as to the perceived relevance of the AAPA in regards to Applicants' circuit of Claim 1 is requested. Moreover, the rejection is tantamount to an improper obviousness-type rejection wherein the Examiner is attempting to piece together the invention from two different prior art architectures which are mutually exclusive and teach away from one another. For at least these reasons, the rejection of Claim 1 is improper, and should be withdrawn.

Claims 6-8 depend from Claim 1 and describe additional aspects related to the circuit. Since no circuit is described in the AAPA as previously discussed, the AAPA most certainly does not teach, or even suggest, these additional claimed aspects. For the reasons discussed in regards to Claim 1, and because the AAPA does not teach or suggest the additional circuit aspects of Claims 6-8, these Claims are allowable over this rejection, which is improper, and should be withdrawn.

Turning next to independent Claim 19, this Claim includes aspects that are similar to those discussed in reference to Claim 1 as follows:

> 19. A system for use in managing requests within a data processing system, comprising:
>
> means for providing data in response to a request before all read-only copies of the data that reside within the data processing system at the time of receipt of the request have been invalidated; and

means for selectively discontinuing predetermined data processing tasks until all of the read-only copies have been invalidated.

This Claim includes means for providing data in response to a request before all read-only copies of the data have been invalidated. Thus, the first prior art architecture described on page 2 of Applicants' Specification does not apply to the invention of Claim 19, since that architecture "...invalidat[es] old copies of data before data is provided to an IP". (Specification page 2 line 23.) Moreover, the description related to this first architecture most certainly does not teach or suggest any means for selectively discontinuing data processing tasks until invalidation is complete, since at the time the data is provided to the IP, invalidation is guaranteed to be complete. No discontinuation of processing is needed. The passage that the Examiner is citing as teaching this aspect is merely stating the obvious: the IP cannot process the data until the data is received. However, this passage most certainly does not teach any means for selectively discontinuing data processing tasks.

To summarize, the AAPA description on page 2 regarding the first prior art architecture is completely unrelated to the invention of Claim 19. In fact, it teaches away from Claim 19 by teaching a system that will not provide data to a requester until all invalidation operations are completed. Moreover, this passage most certainly does not teach Applicants' means for selectively discontinuing data processing task, since processing of the data within that AAPA system cannot possibly even be initiated until that data is received from the memory, at which time the invalidation operations are guaranteed to be completed. For all of the foregoing reasons, the description on page 2 of the AAPA that is related to the first prior art architecture most certainly does not teach or suggest either of the elements of Applicants' Claim 19.

Next, the second cited passage of the AAPA on page 3 concerning the second prior art architecture is considered in regards to Claim 19. According to

this second prior art architecture, data is provided to an IP before invalidation is completed. However, this architecture most certainly does not describe any means for selectively discontinuing predetermined data processing tasks until invalidation of data is complete. Means for discontinuing data processing tasks are not needed to maintain coherency in this second prior art architecture, since coherency is maintained through the use of the dual-channel memory design, and not through interrupting any processing activities, as previously discussed.

To summarize, neither of the prior art architectures set forth in the AAPA describe any means for discontinuing processing activity until invalidation operations are completed within a system that provides data to a requester before invalidation operations are completed. For at least this reason, the rejection is improper. In addition, it is reiterated that the various passages of the AAPA cited in the action in regards to Claim 19 describe two different prior art architectures. The first architecture is based on not providing data prior to the completion of invalidation requests while the second is predicated on providing data prior to the completion of invalidation requests, but only using a very specific type of single-channel memory architecture to do so. The two prior art architectures are mutually exclusive, and the Examiner cannot, therefore, cite a portion of the description related to the first architecture as teaching some aspects of Applicants' invention, while citing portions of the description related to the second architecture as teaching other aspects of Applicants' invention, as the Examiner has done in the rejection of Claim 19. (See Office Action page 4, second full paragraph.) This rejection is tantamount to an obviousness-type rejection wherein the Examiner is combining two different teachings that very specifically teach away from one another. The Examiner's piecing together of the unrelated portions of two diametrically-opposed architectures is not only impermissible, but would yield a system that is non-operational. For all of these reasons, the Examiner's rejection of Claim 19 is improper, and should be withdrawn.

Claims 20-22 depend from Claim 19 and include additional scopes and aspects not taught by the AAPA. For the reasons discussed in regards to Claim 19, and because of the additional aspects that are not taught, these Claims are allowable over the current rejection, which is improper and should be withdrawn.

4. Claims 2 and 9-18 are rejected under 35 USC §103(a) as being unpatentable over AAPA in view of U.S. Patent No. 6,792,507 to Chiou et al. ("Chiou"). This rejection is respectfully traversed.

Claim 2 describes the aspect of the invention wherein data is provided before the invalidation operations are completed. The Examiner states on page 6 of the Office Action that the "AAPA does not specifically disclose the data is provided before the invalidation operations are completed." This statement is confusing in light of the §102 rejection of Claim 19 that is discussed above. Recall Claim 19 includes "means for providing data in response to a request before all read-only copies of the data...have been invalidated". In regards to this element of Claim 19, the Examiner cites AAPA page 3 lines 1-3. (See Office Action page 4, second full paragraph.) If this rejection is maintained, more detailed clarification is requested regarding the perceived distinction between Claim 19 and that of Claim 2.

Turning next to the specifics of this rejection, it may first be noted that absolutely nothing in Chiou or the AAPA, either alone or in combination, teaches or suggests a circuit to execute an instruction that causes the first requester to temporarily enter a stalled state until all invalidation operations have been completed. For this reason alone, the rejection is improper, and Claim 2 is allowable as presently presented.

Additionally, Chiou does not teach the aspect of the invention for which it is cited. This can be observed by considering the Chiou system generally, and the portion of Chiou cited by the Examiner more specifically.

The Chiou system provides a first cache within a network (e.g., the Internet) that is located near some storage devices. This cache is considered the "storage-side" cache. The Chiou system also provides another cache that is located near a requesting host system. This is considered the "host-side" cache. Data that is frequently accessed by a group of host/client systems is cached in the host-side cache. This is said to allow a network of arbitrary size to exist between the host-side and the storage-side caches. (Chiou column 2 lines 58-67.)

According to Chiou, each read request from a host system is first directed to the host-side cache. If that host-side cache cannot satisfy the request, that request is sent to the storage-side cache. If neither cache stores the data, the request is forwarded to a storage device (also referred to in Chiou as the "target storage device", or merely the "target device"). As previously discussed, this target device is part of the main storage facility for the system. (Chiou column 3 lines 1-12.)

Before continuing, it may be appreciated by the foregoing that in Chiou, the "host systems" are the "requesters" that are requesting data, and the "target storage devices" are the main storage facility that correlates to Applicants' memory.

Next, the passage cited by the Examiner is considered. This passage is describing a host write operation wherein a host system (i.e., a requester) writes updated data to the target storage device (corresponding to Applicants' memory, Chiou column 14 lines 56-58.) Several modes of operation exist for performing this type of write operation. In one mode, the updated data is stored back to the target storage device before the data copy within that host's cache is either updated or invalidated. (Chiou column 15 lines 4-8.)

Thus, the cited passage of Chiou is describing

- a requester (i.e., the host) that is
- storing updated write data back

- to the target storage (corresponding to memory) before the data copy is invalidated or updated in the cache that is coupled to the requester.

In contrast, Applicants' Claims 1 & 2 describe

- a memory

- providing requested read data

- to a requester before read-only copies that are stored by one or more other requesters are invalidated. (See Claim 1 lines 3-6.)

Thus, whereas Chiou is providing write data from the requester to the memory, Applicants' Invention is providing read data from the memory to the requester. Whereas Chiou is invalidating data within the requester's cache, Applicants' invention is invalidating data within all other caches except the requester's cache. Based on this comparison, the perceived relevance of Chiou in regards to Applicants' claimed invention of Claims 1 and 2 is not understood. If this rejection is maintained, a more detailed discussion of the relevance of Chiou is respectfully requested.

For at least the above-described reasons, Claim 2 is allowable over the current rejection, which is improper, and should be withdrawn.

Turning now to a discussion of Claims 9-19, Claim 9 is as follows:

> 9. For use in a system having multiple requesters coupled to a shared memory, a method for controlling processing of requests, comprising:
>
> a.) Issuing a request for data by a requester to the shared memory;
>
> b.) providing the data from the shared memory in response to the request before all read-only copies of the data retained by other requesters have been invalidated; and

c.) stalling the requester until all of the read-only copies have been invalidated.

This Claim relates to a system for providing data from the shared memory before all read-only copies of the data have been invalidated. As such, nothing in AAPA page 2, last paragraph that describes the first prior art architecture relates to Claim 19, since that architecture specifically relates only to a system for providing data after all copies have been invalidated.

Additionally, nothing in the AAPA describing the second prior art architecture relates to Claim 19, since that description has absolutely nothing to do with stalling the requester. As previously discussed, that second prior art architecture depends on the single-channel memory design to maintain coherency, and does not require impacting operation of the requester in any way.

Finally, as previously discussed, the cited passage in Chiou does not relate to providing data from a shared memory to a requester before invalidation activities are completed, but instead relates to providing data from a requester to shared storage.

To summarize, nothing in Chiou or the AAPA, alone or in combination, describes a method of providing data before read-only copies of the data are invalidated, and then stalling the requester until all read-only copies have been invalidated. The rejection of Claim 9 is therefore improper, and should be withdrawn.

Claims 10-18 depend from Claim 9, and include other aspects related to providing data before, then stalling the requester until, invalidation activities are complete. These additional aspects are not taught or suggested by the prior art since that art does not teach or suggest even the basic tenets of the invention. For these additional reasons, and for all of the reasons discussed in regards to Claim 9, the rejection of Claims 10-18 is improper and should be withdrawn.

5. Claims 3-5 were rejected under 35 USC §103(a) as being unpatentable over AAPA in view of Chiou and further in view of U.S. Patent No. 6,647,453 to Duncan et al. ("Duncan"). This rejection is respectfully traversed.

Claims 3-5 depend from Claim 1. As already discussed, Claim 1 describes a circuit within a requester that executes an instruction that causes the requester to enter a stalled state until Invalidation operations are completed. Nothing in the AAPA, in Chiou, or in Duncan, describes such a circuit. For at least this reason, Claims 3-5 are allowable over the current rejection, which is improper and should be withdrawn.

Claims 3-5 further describe the aspect wherein the memory is a dual-channel memory having both a request and a response channel. This is said to be taught by Duncan.

Duncan describes an SMP system that has a plurality of processor modules 300, each including several processors. These processor modules 300 are interconnected to form a two dimensional (2D) torus configuration. This SMP system has a plurality of virtual channels including a Request, a Response, and a Forward channel. (Duncan column 6 line 66 – column 7 line 1.)

The portion of Duncan cited by the Examiner states that to load data into its cache, a processor 202 within one of the processor modules may issue any one of several types of requests on the Request channel. (Duncan column 7 lines 47-50.) That request is issued to the directory that identifies the requested data. (Duncan column 7 lines 50-51.) If the requested data is owned by another processor 202, the directory will issue a message to that other processor on the Forward Channel. The other processor, in turn, will acknowledge that is has invalidated its data copy by sending a message back to the directory on the Response channel. (Duncan column 7 lines 55-60.) Thus, the cited passage of Duncan appears to be discussing a symmetrical multiprocessor architecture that connects a plurality of processors to a directory. These processors appear to utilize Request and Response channels to send requests and responses, respectively, to the directory. This type of architecture most certainly does not

teach or in any way suggest any type of a memory circuit that is coupled to multiple requesters, and that includes both request and response channels, as is claimed by Applicants' Claims 3-5. For at least this additional reason, this rejection is improper, and should be withdrawn.

6. The prior art made of record and not relied upon has been reviewed and is considered to be of general interest only.

Conclusion

In the Office Action dated July 18, 2005 ("Office Action"), Claims 1-22 were rejected. In view of the comments set forth above, it is respectfully submitted that all Claims are allowable over the cited prior art, and are in condition for allowance as presently presented. Therefore, a Notice of Allowance is respectfully requested. In the Examiner has questions or comments regarding any of the foregoing, a call to the undersigned is encouraged and welcomed.

Respectfully submitted,

Beth L McMahon 10/13/2005

Beth L. McMahon
Attorney for Applicants
Reg. No. 41,987
Tele No. (651) 635-7893

Unisys Corporation
M.S. 4773
P.O. Box 64942
St. Paul, MN 55164-0942